Table of Contents

Making Your Own Clone Bandits Map

Introduction

This document assumes that you've made unreal levels before so you have prerequisite knowledge about how to use Unreal Ed and know all of the details needed to create a simple multi−player map. Below you will find what additional steps you must take to set up a Clone Bandits map.

Ingredients

The following are the bare essential actors you will need to create a Clone Bandits map

- PlayerStarts for each team
- A BlueCloneJarBase and a RedCloneJarBase
- At least one ClonePump somewhere in the level
- An imported 256x256 Radar Map texture
- RoadPathNodes and good wholesome vehicles to use them
- And of course a level to play in!

Also you will need to set the DefaultGameType to "Clones.ClonesGame"

After you have all of that set up properly, you should be good to go! Here is how to set up each of those things.

Player Starts

Place a few more PlayerStarts in each game than you expect to have people playing. For instance, if you are making a 10−12 person map, be sure you have about 14−16 PlayerStarts (7 to 8 per side).

In the PlayerStart Properties under the PlayerStart tab, you will want to set up the fields as follows:

Property	***Setting***
bCoopStart	False
bEnabled	True
bPrimaryStart	False
bSinglePlayerStart	False
TeamNumber	0 for red / 1 for blue

Clone Jar Bases

Both sides will need at least one CloneJarBase. To place one you will need to load the Clones.u code package from the Actors Browser. Just go to the file menu in the Actors Browser and open the Clones.u file which should be located in your system folder. If you have already loaded a Clone Bandits map, the Clones.u file should already be loaded. To place the Red and Blue CloneJarBase actors navigate through the Actors tree as follows to select and then add by right clicking in the level:

- Actor
    - ♦ NavigationPoint
        - ◊ JumpDest

- JumpSpot
  - GameObjective
    - CloneJarBase
      - BlueCloneJarBase
      - RedCloneJarBase

Within the properties of each CloneJarBase you will be able to set up the following fields:

Property	*Description*	*Default*
FracOfClonesToTake	This sets the fraction of the difference of remaining clones to steal. In other words if this value is set to 0.25 and the Red team has 30 clones and the Blue team has 10 clones, a stolen red jar will be worth 5 clones (barring any max/min restrictions).	0.25
MaxClonesInJar	This caps the upper bound of how many clones can will be contained within one jar. So no matter what value the FracOfClonesToTake is set to, this is the maximum amount the Clone Jar will be worth.	8
MinClonesToTake	This caps the lower bound of how many clones can will be contained within one jar. So no matter what value the FracOfClonesToTake is set to, this is the minimum amount the Clone Jar will be worth.	4
TakenSound	The sound played when a Clone Jar is stolen.	GameSounds.CTFAlarm
TimeForCloneRespawn	Time in seconds between each time a Clone Jar will attempt to respawn. Note that the respawn time is independent of when a clone jar is stolen.	10.0

CloneJarBases can be placed anywhere in the map and in as great a quantity as you would like.

Clone Pumps

To place one you will need to load the Clones.u package from the Actors Browser as described above (in the "Clone Jar Bases" section). Then navigate through the Actors tree as follows to select and then add by right clicking in the level:

- Actor
  - NavigationPoint
    - JumpDest
      - JumpSpot
        - GameObjective
          - DestroyableObjective
            - ClonePump

Within the properties of the Clone Pump you will be able to set up the following fields (although the defaults should work pretty well):

Property	*Description*	*Default*
ActiveSound	This is the sound that the Clone Pump plays when it has been activated	ONSVehicleSounds−S.PowerNode.PwrNodeActive01
bPowered	This value is not used. Leave it at it's default.	True
bStartNeutral	This ensures that the Pump will not belong to either team at the start. If you set this to false, be sure to also set “DefenderTeamIndex” within the GameObjective properties to either 0 or 1 to assign it to the red or blue team respectively.	True
ConstructedSound	The sound played when the Clone Pump finishes constructing	ONSVehicleSounds−S.PowerNode.whooshthunk
ConstructingTime	This is the length of time it takes for the Clone Pump to finish building on its own	30.0
DestroyedSound	The sound that plays once a clone pump has been deactivated.	ONSVehicleSounds−S.PowerNode.PowerCoreExplosion01
HealedSound	The sound that the Clone Pump plays after it has been fully healed by the secondary fire of the link gun.	ONSVehicleSounds−S.PowerNode.PwrNodeBuilt01
HealingSound	The sound that the Clone Pump plays when it is being healed with the secondary fire of the link gun.	ONSVehicleSounds−S.PowerNode.PwrNodeStartBuild03
NeutralSound	The sound played for when the Clone Pump is not controlled or being captured.	ONSVehicleSounds−S.PowerNode.PwrNodeNOTActive01
NumClonesToTake	The amount of Clones the Clone Pump will transfer from one base to the other when it transfers clones.	1
SecondsPerClone		10.0

	The time in seconds between when an activated pump will transfer the specified number of clones from one team to the other.	
StartConstructingSound	The sound played when the Clone Pump is touched while neutral	ONSVehicleSounds−S.PowerNode.PwrNodeBuild02

Clone Pumps can be placed anywhere in the map. If turrets are placed within the value specified under BaseRadius in the GameObjective properties, they will become associated with the team that controls the pump. Thus associated turrets cannot be used by players of the opposing team.

Radar Map

Creating the Texture

You can of course create the texture however you would like, but I find it quick and easy to get a fast start on creating the map by following these quick simple steps:

1. Hide all Actors you do not want visible in the Radar Map from the Perspective view port (O hides volumes, B hides the builder brush, and setting display sizes and using the groups browser can hide most everything else)
2. Find the perspective viewport camera icon in one of the 2D view ports and select it
3. Move the perspective viewport camera high enough so that the entire playing area is viewable when looking down
4. press F4 to bring up the perspective viewport camera's properties
5. Expand the movement tab and set the X and Y portions of the Location to 0 and 0 (assuming that you've centered your map around the origin in Unreal Ed), and under the Rotation portion, set Pitch and Yaw both to −16384 and Roll at 0
6. Press Alt + Print Screen, paste into an image editor, and you have you a fast head start on your Radar Map

Alternatively, you can just take a screen shot of the Top down viewport, but all of the Staticmeshes will show up as green wireframes and the hassle of selecting the Perspective Viewport Camera and setting its rotation and location seems like a much smaller hassle then painting in any StaticMeshes you want to show up in the Radar Map

You may want to add a few more details and enhance some of the colors in your Radar Map image, but once you crop the image to the size of your Terrain,* then everything should just fall into place.

**NOTE: The Radar Map will use the size of the Terrain with the Tag "PrimaryTerrain" to determine what scale to draw the autogenerated portions of the Radar Map –such as the Jar Points and Clone Pumps.*

Assigning the Texture

The mini map is just a texture that is assigned in the Level Properties (accessible from the main Edit menu) under the RadarMap property in the RadarMapImage field. These are the requirements for the Radar Map texture:

- 256 x 256 in size
- saved as a 24 bit Targa file
- Leave all the import settings at their default

For simplicity's sake you can just import the texture into the myLevel package and you can assign it to the RadarMapImage slot by selecting it in the Texture Browser and then pressing the "Use" button in the RadarMapImage field of the Level Properties.

To make sure that the auto−generated portions of the Radar Map (i.e., the Jar Point and Clone Pump locations) show up in the correct places, you will need a Terrain in the level that has its tag set to "PrimaryTerrain" as alluded to earlier. If you do not have a Terrain in your map, the Radar Map will assume that your playing in a 16384 x 16384 sized map.

Vehicles and Paths

To add a vehicle, you will actually want to add a VehicleFactory, otherwise if you just add the vehicle, new vehicles will not respawn and when that placed vehicle is destroyed, it will be gone for the rest of the game. From the Actors Browser expand the following tree to find the Clones Vehicle Factories (note that you may need to load the Clones.U package if you have not done so already –for instructions on how to do this see above (in the "Clone Jar Bases" section).

- Actor
    - ♦ SVehicleFactory
        - ◊ ClonesVehicleFactory
            - · HotRodFactory
            - · LoungeTankFactory
            - · RocketBikeFactory

Then to place them in the level, simply right click and choose the 'Add *CloneVehicle* Here' option.

Once you've placed a Vehicle Factory in your level, there will be two sets of properties that you can change, ClonesVehicleFactory and SVehicleFactory. Below are descriptions of each of the properties in these sections.

ClonesVehicleFactory

Property	***Description***	***Default***
RespawnTime	The time in seconds from when a person leaves with the current vehicle spawned until when the next vehicle will spawn. Note that as soon as someone enters the vehicle, the timer for the next vehicle to spawn in will start.	15.0
TeamNum	Leave this field at its default setting as it is not used.	0

SVehicleFactory

Property	***Description***	***Default***
MaxVehicleCount	This sets the maximum number of vehicles that the spawner will allow in the map. With the default setting, if there are 3 vehicles produced from this factory in the map, then it will not spawn any more until one of those 3 are destroyed.	3

VehicleClass	This defines what vehicle the spawner will produce. Leave it at its default setting	Clones. *VehicleName*

Next (if you want bots to be able to use your map) you will need to add both regular old school PathNodes as well as RoadPathNodes. To place a RoadPathNode, open up your Actors Browser and follow through the Actors tree like so to find the RoadPathNode actor.

- Actor
  - ♦ NavigationPoint
    - ◊ PathNode
      - · RoadPathNode

RoadPathNodes do not need to be placed as close to other RoadPathNodes as regular PathNodes do, but after rebuilding if you find that your paths aren’t connecting as well as you would like, you can use the ForcedPaths array to force certain connections (up to four) or use the ProscribedPaths array to deny certain paths (up to four). These arrays can be found within the NavigtionPoint section of the RoadPathNode properties. Not that when you use these arrays you must use the RoadPathNodes’s unique name found in the Object section of its properties.